正课:

1. 事件:

2. 动画:

3. 类数组对象操作:

4. 添加自定义函数:

1. 事件:

1. 页面加载后自动执行: 2种:

1. DOMContentLoaded: 仅当dom内容(html和js)加载完就提前触发：

$(document).ready(function(){...})

简写: $().ready(function(){...})

更简化: $(function(){...}) => $(()=>{...})

2. 等待网页所有内容(html,css,js,图片)加载完成，才触发: 晚

window.onload=function(){...}

2. 鼠标事件:

mouseover mouseout : 进出子元素，同样反复触发父元素的处理函数

mouseenter mouseleave : 进出子元素，不再反复触发父元素的处理函数

hover等于同时绑定鼠标进入和移出事件

hover(鼠标进入, 鼠标移出)

强调: 代替的是: mouseenter和mouseleave

其实可以更简化:

如果鼠标进入或移出事件，可统一为一个函数，只需要传入一个函数即可。即响应鼠标进入，又响应鼠标移出事件

3. 模拟触发:

$(...).事件名() 调用该处理函数

$(...).trigger("事件名")

2. 动画:

简单动画: 3种:

1. 显示隐藏: .show() .hide() .toggle()

说明: 没有参数时，默认使用display控制显示隐藏，所以，不带动画效果

带动画效果: .show(ms)

问题:

1. 效率: 用定时器实现的动画效果，效率低

2. 效果是写死的，不便于维护

解决: 用transtion 代替 动画API

2. 上滑下滑: .slideUp() slideDown() slideToggle()

3. 淡入淡出: .fadeIn() fadeOut() fadeToggle()

万能动画:

$(...).animate({

目标样式

},ms)

强调: animate只支持单个数值的属性

问题:

1. 不支持颜色动画

2. 用定时器实现的动画效果

排队和并发:

并发: 多个css属性同时变化

放在一个animate内的多个css属性，是并发变化

排队: 多个动画效果，先后顺序执行

对同一个元素，先后调用多个动画API，则默认是排队执行

动画结束后自动执行:

$(...).动画API(......, callback)

callback会在动画结束后，自动执行

callback中的this，自动指当前正在播放动画的DOM对象

停止动画:

$(...).stop()

问题: 默认仅停止队列中当前正在播放的一个动画!

解决: 停止当前动画并清空队列: .stop(true)

选择/判断一个元素是否正在播放动画:

:animated

3. 类数组对象操作:

$(...).each( (i,elem)=>{ .forEach()

elem->当前正在遍历的元素->DOM

对elem执行任何操作

})

鄙视: $(...).each() vs $.each(obj, fn)

1. 存储位置:

$(...).each() 保存在jQuery类型的原型对象中

$.each() 定义在jQuery构造函数上

2. 调用方式:

$(...).each() 只能用jQuery类型的子对象调用

$.each() 让任何类数组对象都可享用each

$(...).index() indexOf()

$("选择器").index(DOM/jQ)

在左边的结果集合中，找右边的DOM或jQ元素的位置i

$("选择器").index("选择器")

在右边的结果集合中，找左边的选择器对应元素的位置i

$("选择器").index() 在当前元素的父元素内，找当前元素的位置i

4. 添加自定义函数:

如果jQuery的API不够用时:

如何添加:

向jQuery.fn中添加新函数=function(){  
 //this->将来调用该函数的jQuery类型的子对象  
 }

强调: 向jQuery原型对象中添加新函数，首先要验证是否已经引入了jQuery类型